EECS 3401, Assignment 2 February 28, 2018

PROBLEM 1: Find the resolvents of the following pairs of clauses (if possible):

```
(a) set(11,Day,next(g(Time))):-
    and
    :-set(Month,time(gmt,Year),next(Year))
(b) add(X,bonus(Z),bonus(X)):-
    and
    :-add(a,bonus(X),bonus(Z)), sal(a,X,Y,Z), add(earn(a,Y),bonus(b),bonus(Y))
(c) p(add(Y,11),W,sq(B)):- q(A,V,add(W,10),add(B,9))
    and
    :-p(A,A,V), q(A,A,B,B)
```

Use PROLOG's = predicate to determine whether two literals are unifiable or not.

PROBLEM 2: Write a PROLOG program that, given two lists L1 and L2, determines which of the lists is longer. In your implementation, use the predicate longer(L1,L2,L) intuitively defined as follows:

```
    longer(L1,L2,L) is true when L is the longest list among L1 and L2
    (if L1 and L2 are of the same length, then L can be either L1 or L2).
```

Show all the resolution steps (including unifiers) taken by PROLOG to answer the query:

```
        :- longer([2],[a,b,c],L).
```

PROBLEM 3: Write a PROLOG program to replace an item X with Y in a list L. In your implementation, use the predicate replace(X,Y,L,Result) intuitively defined as follows

```
 replace(X,Y,L,Result) is true iff Result is the list obtained from List
                       by replacing every element X in L with Y.
```

So, the query

```
        :-replace(3,b,[[3],3,b],R).
```

should be answered

```
        R= [[3],b,b].
```

Show all the resolution steps (including unifiers) taken by PROLOG to answer the query:

```
:-replace(3,b,[[3],3,b],R).
```

PROBLEM 4: Your "replace" program from PROBLEM 3 will not replace items recursively, i.e.,

```
:-replace(3,b,[[3],3,b],R).
```

is answered

```
R= [[3],b,b].
```

rather than

```
R= [[b],b,b].
```

This time you task is to write a PROLOG program which recursively replaces every occurrence of item X with Y. Test your program on a few examples and include the results in your assignment.

PROBLEM 5: Write a PROLOG program to replace an item X with Y in a binary tree T. In your implementation, use predicate

rt(X,Y,T,New_T)

where X is to replace every element Y in the tree T, and New_T is the resulting tree. So, the query

```
:-replace(6,b,t(t(nil,a,nil),b,t(nil,6,nil)),New_T).
```

should be answered

```
New_T=t(t(nil,a,nil),b,t(nil,b,nil))).
```

Fully test your program and include the test results in your report.

PROBLEM 6: Write a PROLOG program that given a binary tree creates the list of all the nodes of the tree. In your implementation, use predicate

t2l(T,L) which is true if L is the list of all elements of of a tree T.

So, the query

```
:-t2l(t(t(nil,3,nil),5,t(nil,2,nil)),L).
```

should be answered

```
L=[3,5,2].
```

Fully test your program and include the test results in your report.

PROBLEM 7: Write a PROLOG program that given a list L returns a binary search tree whose nodes are exactly the elements of L.